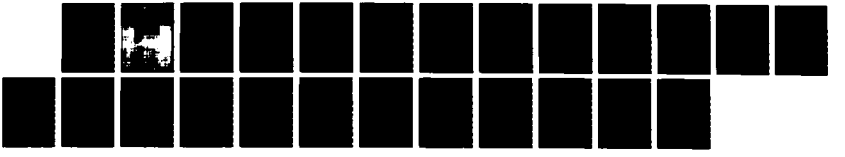
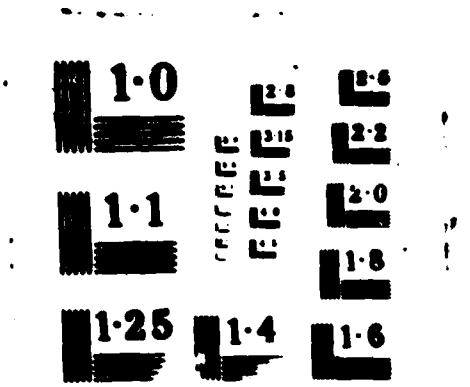
AD-A170 477 ODRPACK SOFTWARE FOR WEIGHTED ORTHOGONAL DISTANCE REGRESSION(U) COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE P T BOGGS ET AL. FEB 87 CU-CS-360-87 1/1

UNCLASSIFIED ARO-21453.7-MA DAAG29-84-K-0140 F/G 9/2 NL




1·0
1·1
1·25
1·4
2·2
2·0
1·8
1·6

UNIVERSITY OF COLORADO

ODRPACK
SOFTWARE FOR WEIGHTED ORTHOGONAL DISTANCE REGRESSION

Paul T. Boggs*
Richard H. Byrd**
Janet R. Donaldson***
Robert B. Schnabel****

CU-CS-360-87 February 1987

DEPARTMENT OF COMPUTER SCIENCE
CAMPUS BOX 430
UNIVERSITY OF COLORADO, BOULDER
BOULDER, COLORADO 80309-0430

Technical Report

2

ODRPACK
SOFTWARE FOR WEIGHTED ORTHOGONAL DISTANCE REGRESSION

Paul T. Boggs*
Richard H. Byrd**
Janet R. Donaldson***
Robert B. Schnabel****

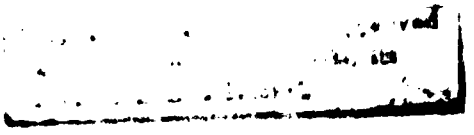
CU-CS-360-87 February 1987

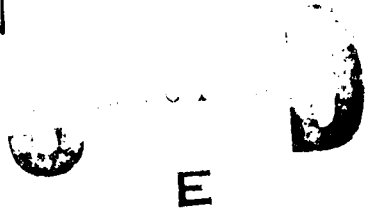
* Optimization Group/Scientific Computing Division, National Bureau of Standards, Gaithersburg, MD 20899

** Department of Computer Science, University of Colorado, Boulder, CO 80309
(Research supported in part under National Science Foundation Grant DCR-8403483)

*** Optimization Group/Scientific Computing Division, National Bureau of Standards, Boulder, CO 80303-3328

**** Department of Computer Science, University of Colorado, Boulder, CO 80309 and Optimization Group/Scientific Computing Division, National Bureau of Standards, Boulder, CO 80303-3328 (Research supported in part under Army Research Office Contract DAAG 29-84-K-0140)





E

Abstract

In this paper, we describe ODRPACK, a software package for the Orthogonal Distance Regression (ODR) problem. This software implements the algorithm described in [BogB85] for finding the set of parameters that minimize the sum of the squared orthogonal distances from a set of observations to a curve determined by the parameters. It can also be used to solve the ordinary nonlinear least squares problem. The ODR procedure has application to curve fitting, and to the errors in variables problem in statistics. The algorithm implemented is an efficient and stable trust-region (Levenberg-Marquardt) procedure that exploits the structure of the problem so that the computational cost per iteration is equal to that for the same type of algorithm applied to the ordinary nonlinear least squares problem. The package allows a general weighting scheme, provides for finite difference derivatives, and contains extensive error checking and report generating facilities.

keywords
orthogonal distance regression
nonlinear least squares
errors in variables

1 Introduction

In [BogBS85] the authors provide an efficient and stable algorithm for the orthogonal distance regression (ODR) problem. This problem is defined as follows. Let (x_i, y_i), $i = 1, \ldots, n$, be a given set of data where $x_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}^1$. Suppose that the values of y_i are a function of x_i and a set of unknown parameters $\beta \in \mathbb{R}^p$, but that both the y_i and the x_i contain actual, but unknown, errors $\epsilon_i^a \in \mathbb{R}^1$ and $\delta_i^a \in \mathbb{R}^m$, respectively. Then the observed value of y_i satisfies

$$y_i = f(x_i + \delta_i^a; \beta^a) - \epsilon_i^a \qquad i = 1, \ldots, n$$

for some actual but again unknown value β^a. (Note that we have chosen the sign of ϵ_i to be negative for convenience.) The *orthogonal distance regression* problem is to approximate β^a by finding the β for which the sum of the squares of the n orthogonal distances from the curve $f(x; \beta)$ to the n data points is minimised. This is accomplished by the minimisation problem

$$\min_{\beta, \delta, \epsilon} \sum_{i=1}^{n} (\epsilon_i^2 + \delta_i^T \delta_i)$$

subject to the constraints

$$y_i = f(x_i + \delta_i; \beta) - \epsilon_i \qquad i = 1, \ldots, n.$$

Since these constraints are linear in ϵ_i, we can eliminate them and the ϵ_i, thereby obtaining

$$\min_{\beta, \delta} \sum_{i=1}^{n} \left[(f(x_i + \delta_i; \beta) - y_i)^2 + \delta_i^T \delta_i \right]. \tag{1}$$

We obtain the final form of the problem to be solved by allowing a general weighting scheme. Let w_i, $i = 1, \ldots, n$, be a set of non-negative numbers and let $D_i \in \mathbb{R}^{m \times m}$, $i = 1, \ldots, n$, be positive diagonal matrices. Then (1) is generalised to the *weighted orthogonal distance regression* problem

$$\min_{\beta, \delta} \sum_{i=1}^{n} w_i^2 \left[(f(x_i + \delta_i; \beta) - y_i)^2 + \delta_i^T D_i^2 \delta_i \right]. \tag{2}$$

This weighting scheme enables us to apply our procedure to problems where ϵ_i and δ_i have different variances, and to situations where various observations should be weighted differently.

The algorithm developed in [BogBS85] is a trust-region, Levenberg-Marquardt method that exploits the structure of the problem to obtain a procedure that is both stable and efficient. In the case of ordinary least squares (OLS), i.e., where all of the δ_i are assumed to be zero, the trust region, Levenberg-Marquardt method requires $O(np^2)$ operations per iteration. (See, e.g., [DenS83] and [Mor77].) The order of operations per iteration, and the constant for the highest order term, is the same for the algorithm developed in [BogBS85] as for OLS. Note that a straight forward use of an OLS algorithm on (2) would require $O(n(n+p)^2)$ operations per iteration (see [BogBS85]) which is clearly prohibitive for large values of n.

ODRPACK implements the algorithm presented in [BogBS85]. In § 2 we describe ODRPACK in detail and in § 3 we discuss the installation of the package. Appendices A and B give a sample program and its output, respectively. The complete user's reference guide and installation manual are distributed with the package.

2 Package Overview

2.1 Highlights

ODRPACK is a portable collection of ANSI '77 Fortran subprograms for fitting a model to data. It is designed primarily for instances when the independent as well as the dependent variables have significant errors. ODRPACK embodies a highly efficient algorithm for solving the weighted orthogonal distance regression problem. In addition, it can be used to solve the ordinary least squares problem where all of the errors are attributed to the observations of the dependent variable, y_i.

ODRPACK is designed to accommodate many levels of user sophistication and problem difficulty.

- It is easy to use, providing two levels of user-control of the computations, extensive error handling facilities, comprehensive printed

reports, complete online documentation and no size restrictions other than effective machine size.

- The necessary derivatives (Jacobian matrices) are approximated numerically if they are not supplied by the user. In addition, the correctness of user-supplied derivatives can be verified by the derivative checking procedure provided.
- Both weighted and unweighted analysis can be performed.
- Subsets of the unknowns can be treated as constants with their values held fixed at their input values, allowing the user to examine models with fewer parameters without rewriting the model subprogram.
- ODRPACK has a default scaling algorithm that attempts to automatically accommodate poorly scaled problems in which the model parameters and/or unknown errors in the independent variables vary widely in magnitude. Alternatively, the user can supply appropriate scaling values.
- ODRPACK is portable and is easily used with other Fortran subprogram libraries. (See § 3.3.)

The following sections discuss the use of ODRPACK and provide a brief description of ODRPACK's main features.

2.2 Implementation

2.2.1 Algorithm

A full discussion of the ODRPACK algorithm is found in [BogBS85]. Briefly, ODRPACK implements a trust region Levenberg-Marquardt method, using scaling to accommodate problems in which estimated values have widely varying magnitudes. The Jacobian matrices, i.e., the matrices of first partial derivatives of $f(x;\beta)$ with respect to β and x, are computed at every iteration either by finite differences or by a user-supplied subprogram (see § 2.2.3). The iterations are stopped when any one of three stopping criteria are met. Two of these indicate the iterations have converged to a solution.

These are *sum-of-squares convergence*, which indicates that the change in the sum of the squared weighted observational errors is sufficiently small, and *parameter convergence*, which indicates the change in the parameters is sufficiently small. The third stopping criteria is a limit on the number of iterations.

2.2.2 Starting Values

The user must supply starting values for the unknowns being estimated, i.e., for β and δ. Users familiar with the ordinary nonlinear least squares problem are generally aware of the importance of obtaining good starting values for the estimated function parameters. It is equally important here. Good initial approximations can significantly decrease the number of iterations required to find a solution; a poor initial approximation may even prevent a solution from being found at all. Reasonable initial approximations are often available from previous analysis or experiments. When good starting values are not readily available, the user may have to do some preliminary analysis to obtain them [Him70].

Users who do not provide scale information are strongly encouraged not to use zero as an initial approximation for any of the function parameters, β_k, $k = 1, \ldots, p$, since a zero value can result in incorrect scaling (see § 2.2.4). Setting the initial approximation to the largest magnitude which, for the user's problem, is effectively zero, rather than the actual value zero, will help to eliminate scaling problems and possibly produce faster convergence. For example, if β_1 represents change in a cost measured in millions of dollars, then the value 10 might be considered "effectively zero" and an initial value of $\beta_1^0 = 10$ is preferable to $\beta_1^0 = 0$.

When using orthogonal distance regression it is also important to have good starting values for the estimated errors, δ_i, $i = 1, \ldots, n$. The ODRPACK default is to initialize δ_i to zero, which is the most obvious initial value. (Note that zero starting values for δ_i do not cause scaling problems similar to those discussed above for β.) Initializing δ_i to zero, however, is equivalent to initially assigning all of the errors to the dependent variable as is done for ordinary least squares. While this is quite adequate in many cases, in others it is not. A plot of the observed data and of the curve

described by the model function for the initial parameters may indicate whether or not zero starting values for δ_i are reasonable and may make it possible to visually determine better starting values for some of the δ_i. For example, if such a plot shows that the vertical distance from a data point (x_i, y_i) to the curve is far larger than the orthogonal distance, then δ_i should probably not be initialized to zero. This may occur near an asymptote or near a local minimum or maximum. In such cases, it is often appropriate to initialize δ_i to the horizontal distance from the data point to the curve.

2.2.3 Derivative Handling

As was noted in § 2.2.1 the Jacobian matrices, i.e., the matrices of first partial derivatives of $f(x;\beta)$ with respect to each β and each component of x_i, are required at every iteration. The user may provide the necessary derivatives, or they can be computed by ODRPACK using finite differences. Finite difference derivatives generally cause very little change in the results from those obtained using analytic derivatives, provided sufficient precision is used (typically double precision on a 32-bit machine).

Because coding errors are a common problem with user-supplied derivatives, ODRPACK has an option to check the validity of the user-supplied derivative code by comparing its results to finite difference values for the derivative. The derivative checking procedure examines only one row of the Jacobian matrices, and is therefore quite efficient. Checking only one row is reasonable for regression models since the same code is frequently used to compute the model function and derivatives for each row, as in the case of the example shown in Appendix A.

When the value of the user-supplied derivative disagrees with the corresponding finite difference value, the checking procedure attempts to determine whether the disagreement is due to an error in the user's code or is due to the inaccuracy of the finite difference approximation. The checking procedure automatically generates an error report when one or more of the derivatives are found to be either incorrect or questionable. This report identifies which derivatives appear correct, which appear incorrect, and which appear questionable and for what reason.

2.2.4 Scaling

Poorly scaled problems, i.e., problems in which the unknowns, β and δ, vary over several orders of magnitude, can cause difficulty for least squares procedures. ODRPACK's scaling algorithm attempts to automatically overcome these difficulties, although it is preferable for the user to choose the units of the variable space so that each of the parameters β will have roughly the same magnitude [DenS83]. When the variables have roughly the same magnitude, the ODRPACK scaling algorithm will select scale values which are equal, and the resulting computations will be the same (except for the effect of finite precision arithmetic) as an unscaled analysis, i.e., an analysis in which all of the scale values are set to one. If the variables do not have roughly the same magnitude, the ODRPACK scaling algorithm will select varying scale values. This will not change the optimal solution but may affect the number of iterations required to find the solution, and, in some cases, whether the algorithm is or is not successful.

Users may also select their own scale values as discussed in the ODRPACK reference guide.

2.2.5 Default Values and Structured Arguments

ODRPACK uses default values and structured arguments to simplify the user interface. The availability of default values in ODRPACK means that the user does not have to be concerned with determining values for many of the ODRPACK arguments unless the problem being solved requires the use of non-default values. Structured arguments, described below, can reduce the amount of storage space required for some arguments and the work required by the user to initialize those arguments.

Default Values. Default values have been specified for ODRPACK subprogram arguments wherever feasible. These default values are invoked by setting the corresponding argument to any negative value. Arrays with default values are invoked by setting the first element of the array to a negative value, in which case only the first value of the array will ever be used. This allows a scalar to be used to invoke the default values of arrays, thus saving space and eliminating the need to declare such arrays.

Users are encouraged to invoke the default values of arguments wherever possible. These values have been found to be reasonable for a wide class of problems. Their use will greatly simplify the initial use of ODRPACK for a given problem. Fine tuning of these arguments can then be done later if necessary.

Structured Arguments. Certain ODRPACK arguments specify attributes of the individual components of X and Δ, where X is the $n \times m$ array with i–th row x_i and Δ is the corresponding $n \times m$ array with i–th row δ_i. These attribute arrays are frequently either constant for all components or are constant within each column and vary only among the columns. The *structured argument* facility allows a scalar to specify an attribute of an entire column or of the whole array.

For example, one such attribute array which might display this structure is the matrix of δ weights, RHO, where RHO is an $n \times m$ array with i–th row the diagonal elements of D_i (cf. (2)). Suppose each row of X indicates an hourly temperature reading and each column a different day on which the temperature readings were taken. Then the user would probably want to weight each component of Δ equally, and thus RHO would be constant throughout. If one column of the independent variable contained hourly temperature readings and the other hourly humidity readings, then the user might want to weight each component in the first column of Δ the same, and to weight each component in the second column the same, but not necessarily want to weight the two columns equally. In this case, the components of RHO would be constant within each column and would vary only among the columns. Of course, in other cases, the user might want to weight each component of Δ differently, and each component of RHO would be different.

ODRPACK structured arguments exploit this structure. If each of the $n \times m$ elements of an attribute array is the same, then a single value can be used to specify all $n \times m$ elements. If the values of such an array only vary among the columns, then each column of the array can be specified by a single value. Thus, it is only necessary to supply all $n \times m$ elements of the structured argument array when the elements of one or more of the columns must be individually specified.

2.2.6 Calling Sequences

As we noted in § 2.1, ODRPACK has two levels of user-control of the computations. These levels are provided by user-callable subprograms SODR and SODRC in single precision, and DODR and DODRC in double precision. SODR and DODR have shorter call statements than SODRC and DODRC because they preset many variables to their default values in order to reduce user input. (See § 2.2.5.) We believe that SODR and DODR are adequate for most problems. SODRC and DODRC, with their expanded call statements, provide the user with maximum flexibility in finding the weighted orthogonal distance solution.

2.2.7 Automatic Output

ODRPACK automatically generates computation reports, thus saving the user from the tedious chore of formatting the results for output. The *ODRPACK computation reports* are divided into three parts: an initial summary, iteration summaries and a final summary. Each part can be generated in either a long or short form, and the frequency of the iteration summaries can be specified. By default, the computation reports include a "long" initial summary, no iteration summaries and a "short" final summary, as shown in Appendix B. ODRPACK user-callable subprograms SODR and DODR automatically generate the default report; subprograms SODRC and DODRC allow the user to control the computation reports.

2.3 Example Program

Appendix A shows an example of a user-supplied driver for invoking ODRPACK. This sample program uses the ODRPACK double precision subprogram DODR to solve exercise I on pages 521 - 522 of [DraS81]. (An example using ODRPACK single precision subprogram SODR could easily be generated from the one shown in Appendix A by substituting SODR for DODR and changing the DOUBLE PRECISION declaration statement to REAL.) The model for this example is

$$y_i = \exp\left\{-\beta_1(x_{1,i} + \delta_{1,i})\exp\left[-\beta_2\left(\frac{1}{x_{2,i} + \delta_{2,i}} - \frac{1}{620}\right)\right]\right\} - \epsilon_i \ .$$

The report generated by ODRPACK for this example is shown in Appendix B.

3 Installing ODRPACK

3.1 Supplied Software

3.1.1 ODRPACK Code

ODRPACK software is available in both single and double precision versions. These are distinguished by the first letter of the subprogram name: D for double precision, S for single precision. The code for each version is separated into three sections to facilitate installation.

The first section includes all subprograms written especially for ODRPACK. The two user-callable ODRPACK subprograms of each version are listed first, followed by the ODRPACK exercise subprogram. The remaining subprograms are then listed in alphabetical order. The code in this section should not require any modification, unless the installer wishes to customize the user-callable subprograms as discussed in § 3.2.

The second section of code includes the subprograms used by ODRPACK from the public domain packages LINPACK [DonMBS79] and BLAS [LawHKK79], also listed in alphabetical order. The installer can use local versions of these packages if available. This would be particularly beneficial if the installer's machine has specially optimized versions of LINPACK or BLAS.

The third section includes the only machine dependent subprogram in ODRPACK. This subprogram supplies the machine dependent constants used by ODRPACK. It already includes comment statements listing the necessary constants for a number of common machines. If the constants for the target machine are included in this subprogram, then the installer need only "uncomment" the appropriate DATA statements. Note that this subprogram will return an undefined value until it is updated; the installer *must* update it before compiling and running ODRPACK.

3.1.2 Test Drivers and Data

The ODRPACK supplied software includes drivers and data sets for running ODRPACK in both single and double precision. There are three drivers for each version of the code. Two of these drivers are simple programs that users can modify to form their own ODRPACK drivers. The third is a demonstration program which exercises ODRPACK's main features and can be used to verify that the installation was completed successfully.

The demonstration program calls ODRPACK several times, with each call testing one or more features of the package. The results of each call are automatically compared to the results obtained by the authors using the double precision version of ODRPACK run on a CYBER 855 under NOS 2.4.2 (120 bits per double precision value). The success, or failure, of each test is noted individually in the output generated by this program, and is summarized for all of the data sets. By running this demonstration, the installer can easily be assured that the package is performing as it should.

3.2 Customizing ODRPACK

ODRPACK is designed to easily accommodate a wide variety of problems without any modification to the supplied code. However, installers might want to customize the user interface, i.e., the user-callable subprograms, or to customize the ODRPACK generated reports for specific tasks.

Customizing ODRPACK is simplified by the extensive comments contained within each subprogram, and the consistent use of variable names between subprograms. The SLATEC Source File Format [FonJS82] has been followed, and each subprogram provides a standardized prologue describing the purpose of the subprogram and what other subprograms are called, an alphabetical list of all variables referenced by the subprogram and how they are used, as well as comments explaining the major sections of the code.

In addition, ODRPACK's automatically generated reports are produced by separate subprograms. Because the report generators are isolated in this manner, it is relatively easy to produce customized reports if required.

3.3 Portability

ODRPACK code conforms to the ANSI X3.9-1978 FORTRAN standard and has been successfully installed on a Cyber 855, Cyber 205, IBM PC/AT, Concurrent 3230 and Vax 11/780. We believe it will be possible to install ODRPACK on any system with an ANSI Fortran '77 compiler and adequate memory.

A Sample program

```
      PROGRAM SAMPLE
C
C  SET PARAMETERS FOR MAXIMUM PROBLEM SIZE HANDLED BY THIS DRIVER
C  WHERE  MAXN IS THE MAXIMUM NUMBER OF OBSERVATIONS ALLOWED,
C         MAXM IS THE MAXIMUM NUMBER OF COLUMNS IN THE
C              INDEPENDENT VARIABLE ALLOWED, AND
C        MAXNP IS THE MAXIMUM NUMBER OF FUNCTION PARAMETERS
C              ALLOWED.
C
      INTEGER MAXN,MAXM,MAXNP
      PARAMETER (MAXN=15,MAXM=5,MAXNP=5)
C
C  SET PARAMETERS TO SPECIFY DIMENSIONS OF ARRAYS USED BY DODR
C
      INTEGER LDX,LDRNO,LWORK,LIWORK
      PARAMETER
     +   (LDX=MAXN,
     +   LDRNO=1,
     +   LWORK = 9 + 8*MAXN + 10*MAXN*MAXM + 2*MAXN*MAXNP + 8*MAXNP,
     +   LIWORK = 2*MAXNP + MAXM + 10)
C
C  DECLARE USER-SUPPLIED SUBROUTINES AND
C  ALL OTHER NECESSARY VARIABLES AND ARRAYS
C
      EXTERNAL
     +   FUN,JAC
      INTEGER
     +   N,M,NP,
     +   JOB,
     +   IWORK(LIWORK),
     +   INFO
      DOUBLE PRECISION
     +   X(LDX,MAXM),
     +   Y(LDX),
     +   BETA(MAXNP),
     +   RNO(LDRNO,MAXM),
     +   WORK(LWORK)
```

```
C
      OPEN(UNIT=5,FILE='DATA1')
      OPEN(UNIT=6,FILE='REPORT')
C
C  READ NUMBER OF OBSERVATIONS
C       NUMBER OF COLUMNS OF DATA IN THE INDEPENDENT VARIABLE
C       NUMBER OF PARAMETERS
C       OBSERVED VALUES OF INDEPENDENT AND DEPENDENT VARIABLES
C       STARTING VALUES OF FUNCTION PARAMETERS
C
      READ (5,*) N,M,NP
      READ (5,*) ((X(I,J),I=1,N),J=1,M)
      READ (5,*) (Y(I),I=1,N)
      READ (5,*) (BETA(I),I=1,NP)
C
C  SPECIFY DELTA WEIGHTS
C
      RHO(1,1) = 3.0D0
      RHO(1,2) = 5.0D0
C
C  SET JOB TO DEFAULT SETTING, INDICATING
C       - SOLUTION TO BE FOUND BY ODR
C       - DERIVATIVES TO BE COMPUTED BY FINITE DIFFERENCES
C       - DELTA'S TO BE INITIALIZED TO ZERO
C
      JOB = -1
C
C  COMPUTE ODR SOLUTION USING FINITE-DIFFERENCE DERIVATIVES
C
      CALL DODR
     +    (FUN,JAC,
     +    N,M,NP,
     +    X,LDX,
     +    Y,
     +    BETA,
     +    RHO,LDRHO,
     +    JOB,
     +    WORK,LWORK,IWORK,LIWORK,
     +    INFO)
```

```
C
      END
      SUBROUTINE FUN(N,NP,M,BETA,XPLUSD,LDXPD,F,IFLAG)
C
C  INPUT ARGUMENTS
C  (WHICH MUST NOT BE CHANGED BY THIS ROUTINE)
C
      INTEGER N,NP,M,LDXPD
      DOUBLE PRECISION BETA(NP),XPLUSD(LDXPD,M)
C
C  OUTPUT ARGUMENTS
C
      DOUBLE PRECISION F(N)
      INTEGER IFLAG
C
      DO 10 I = 1, N
         F(I) = EXP(-BETA(1)*XPLUSD(I,1)*
     +           EXP(-BETA(2)*
     +                (1.0D0/XPLUSD(I,2) - 1.0D0/620.0D0)))
   10 CONTINUE
C
      RETURN
      END
```

B Sample output

Report generated by DODR example program from Appendix A, run using a Cyber 180/855 under NOS 2.4.2.

```
*******************************************************
* ODRPACK VERSION 1.2 OF 12-15-86 (DOUBLE PRECISION) *
*******************************************************
```

```
INITIAL SUMMARY FOR FIT BY METHOD OF ODR
========================================
```

```
PROBLEM SIZE:
-------------

NUMBER OF OBSERVATIONS                                  8
NUMBER OF COLUMNS OF DATA IN INDEPENDENT VARIABLE       2
NUMBER OF FUNCTION PARAMETERS                           2
NUMBER OF UNFIXED FUNCTION PARAMETERS                   2
```

```
INDEPENDENT VARIABLE AND DELTA WEIGHT SUMMARY:
----------------------------------------------

                          COLUMN   1                COLUMN   2
                      OBS 1          OBS N      OBS 1          OBS N
          X -    .109000D+03    .680000D+02    .600000D+03    .640000D+03
      FIXED -             NO             NO             NO             NO
INITIAL DELTA -    .000000D+00    .000000D+00    .000000D+00    .000000D+00
```

```
  DELTA SCALE -     .78740D-03    .78740D-03    .15625D-02    .15625D-02
DELTA WEIGHTS -     .30000D+01    .30000D+01    .50000D+01    .50000D+01
```

DEPENDENT VARIABLE AND OBSERVATIONAL ERROR WEIGHT SUMMARY:

```
                          OBS 1         OBS N
                Y -     .91200D+00    .37600D+00
OBS. ERROR WTS. -       .10000D+01    .10000D+01
```

FUNCTION PARAMETER SUMMARY:

```
       INDEX -                  1                 2
INITIAL BETA -     .11550000D-01     .50000000D+04
       FIXED -                 NO                NO
  BETA SCALE -     .96580067D+02     .20000000D-03
```

CONTROL VALUES AND STOPPING CRITERIA:

```
   JOB     TAUFAC      SSTOL     PARTOL  MAXIT
-0001     .10D+01     .50D-14     .96D-19     50
```

A. FIT IS NOT A RESTART.
B. DELTAS ARE INITIALIZED TO ZERO.
C. DERIVATIVES ARE COMPUTED BY FINITE DIFFERENCES.
D. FIT IS BY METHOD OF ORTHOGONAL DISTANCE REGRESSION.

INITIAL SUMS OF SQUARES:

SUM OF SQUARED WEIGHTED OBSERVATIONAL ERRORS	.67662011D+00
SUM OF SQUARED WEIGHTED DELTAS	.00000000D+00
SUM OF SQUARED WEIGHTED EPSILONS	.67662011D+00

FINAL SUMMARY FOR FIT BY METHOD OF ODR
==========

STOPPING CONDITION (INFO = 1):

THE RELATIVE CHANGE IN THE SUM OF THE SQUARED
WEIGHTED OBSERVATIONAL ERRORS IS LESS THAN SSTOL

NUMBER OF ITERATIONS	NUMBER OF FN EVALS	CONDITION NUMBER (INVERSE)	RANK DEFICIENCY
6	38	.1888D-06	0

FINAL SUMS OF SQUARES:

SUM OF SQUARED WEIGHTED OBSERVATIONAL ERRORS	.753823230-03
SUM OF SQUARED WEIGHTED DELTAS	.235420990-07
SUM OF SQUARED WEIGHTED EPSILONS	.753799690-03

ESTIMATED BETA(I), I = 1, ..., NP:

INDEX	VALUE ---------->	
1 TO 2	.36579727D-02	.27627327D+06

ESTIMATED EPSILON(I) AND DELTA(I,*), I = 1, ..., N:

I	EPSILON(I)	DELTA(I,1)	DELTA(I,2)
1	.16752445D-02	.14086172D-06	.42418798D-06
2	.20434718D-02	.12838222D-05	.20262810D-05
3	-.20690065D-01	-.71652291D-06	-.23568824D-04
4	.24305832D-02	.15047092D-05	.24114481D-05
5	.72777482D-02	.23393261D-06	.62079313D-05
6	.40793264D-02	.24162846D-05	.40483552D-05
7	.13043071D-01	.43337344D-06	.14726727D-04
8	-.85499649D-02	-.51394680D-05	-.84861063D-05

References

[BogBS85] Boggs, P. T., R. H. Byrd, and R. B. Schnabel (1985), "A stable and efficient algorithm for nonlinear orthogonal distance regression," University of Colorado Department of Computer Science Technical Report Number CU-CS-317-85. (To appear in *SIAM J. Sci. Stat. Computing*, 1987.)

[DenS83] Dennis, J. E., and R. B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.

[DonMBS79] Dongarra, J. J., C. B. Moler, J. R. Bunch, and G. W. Stewart (1979), *LINPACK Users' Guide*, SIAM, Philadelphia, PA.

[DraS81] Draper, N. R., and H. Smith (1981), *Applied Regression Analysis, Second Edition*, John Wiley and Sons, New York, NY.

[FonJS82] Fong, K. W., T. H. Jefferson, T. Suyehiro (1982), "SLATEC common mathematical library source file format", Lawrence Livermore Laboratory UCRL-53313.

[Him70] Himmelblau, D. M. (1970), *Process Analysis by Statistical Methods*, John Wiley and Sons, New York, NY.

[LawHKK79] Lawson, C., R. Hanson, D. Kincaid, and F. Krogh (1979), "Basic linear algebra subprograms for FORTRAN usage", *ACM Trans. Math. Software*, 5(3):308-371.

[Mor77] Moré, J. J. (1977), "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical Analysis*, G. A. Watson, ed., Lecture Notes in Math. 630, Springer Verlag, Berlin, 105-116.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARO 21453.7-MA	2. GOVT ACCESSION NO. N/A AD-A178 477	3. RECIPIENT'S CATALOG NUMBER N/A
4. TITLE (and Subtitle) ODRPACK-Software For Weighted Orthogonal Distance Regression		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Paul T. Boggs Robert B. Schnabel Richard H. Byrd Janet R. Donaldson		8. CONTRACT OR GRANT NUMBER(s) DAAG-29-84-K-0140
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE February 1987 13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Orthogonal Distance Regression, Nonlinear Least Squares, Errors in Variables

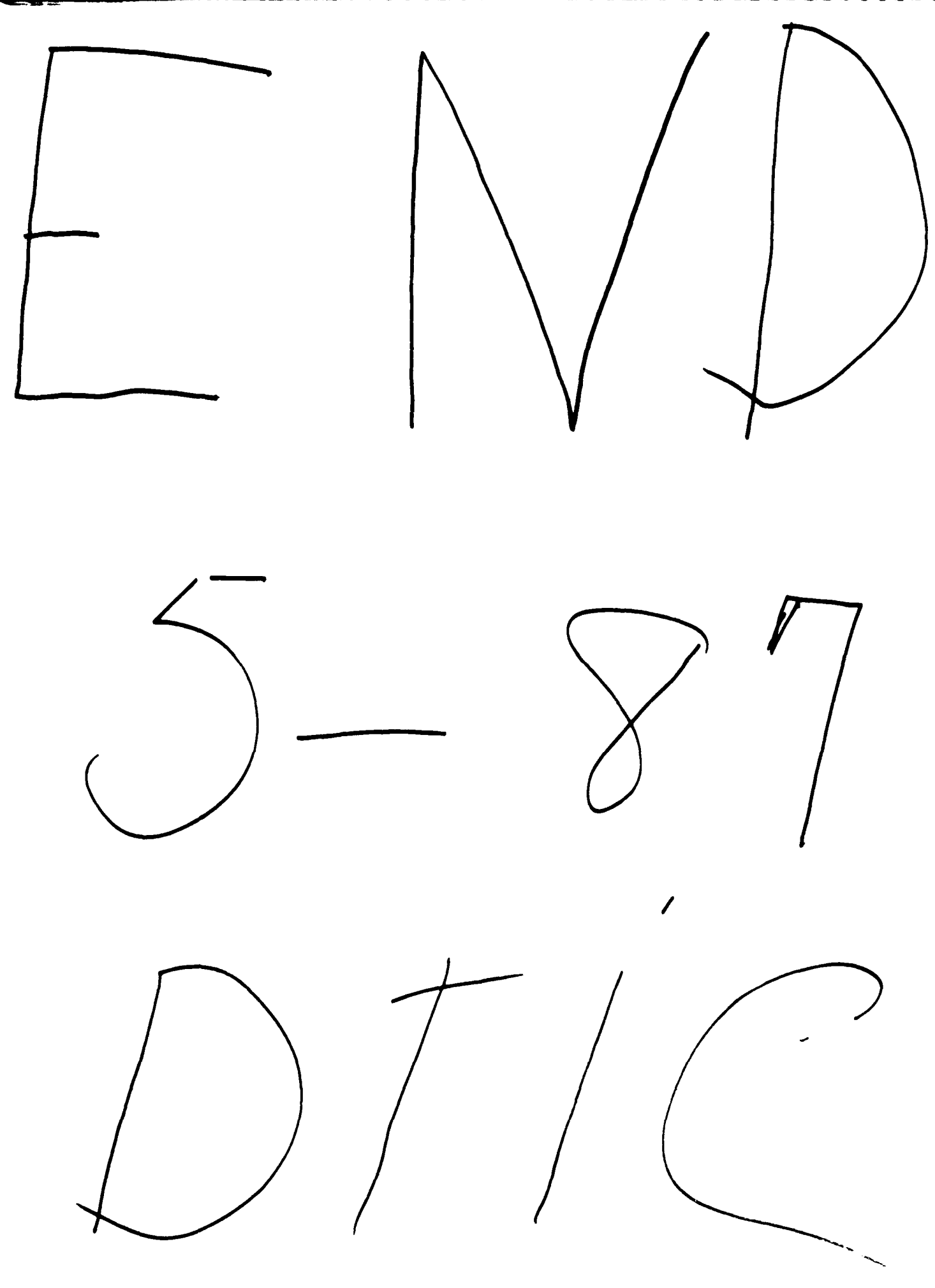
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Attached

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)


E N D
5 — 87
D T I C